Markov Decision Processes and Policy Iteration

Last Time

- What does "**Markov**" mean in "Markov Process"?
- What is a **Markov decision process**?

Guiding Questions

- What is a **Markov decision process**?
- What is a **policy**?
- How do we **evaluate** policies?

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1,2,3\}$ $(x,y) \in \mathbb{R}^2$ $\{0,1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s,i,r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1,2,3\}$ $\mathbb{R}^2$ $\{0,1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

$T(s' \mid s, a)$

- R (reward function) - maps each state and action to a reward

$R(s,a)$ or $R(s,a,s')$

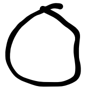
- γ: discount factor

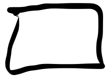
$s', r = G(s,a)$

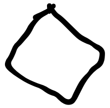
- b: initial state distribution
- S_t: set of terminal states

Decision Networks and MDPs

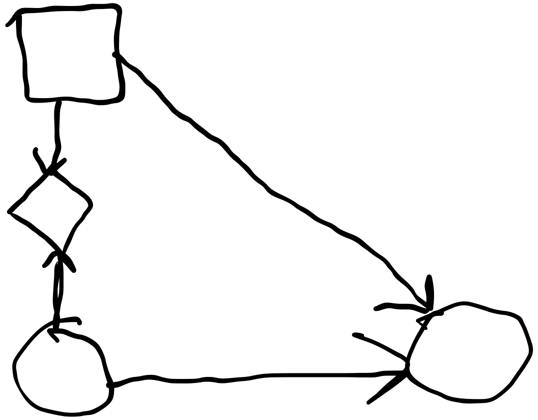
Decision Network

 Chance node

 Decision node

 Utility node

MDP Dynamic Decision Network



MDP Example

Imagine it's a cold day and you're ready to go to work. You have to decide whether to bike or drive.

- If you drive, you will have to pay $15 for parking; biking is free.
- On 1% of cold days, the ground is covered in ice and you will crash if you bike, but you can't discover this until you start riding. After your crash, you limp home with pain equivalent to losing $100.

Policies and Simulation

- A *policy*, denoted with $\pi(a_t \mid s_t)$, is a conditional distribution of actions given states.
- $a_t = \pi(s_t)$ is used as shorthand when a policy is deterministic.
- When a policy is combined with an MDP, it becomes a Markov stochastic process with

$$P(s' \mid s) = \sum_a T(s' \mid s, a)\, \pi(a \mid s)$$

MDP Objective:

$$U(\pi) = \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi\right]$$

<u>Algorithm: Rollout Simulation</u>

Inputs: MDP (S, A, R, T, γ, b) (only need generative model, G), Policy π, horizon H

Outputs: Utility estimate $\hat{u}$

$s \leftarrow \mathrm{sample}(b)$

$\hat{u} \leftarrow 0$

for t in $0 \ldots H-1$

    $a \leftarrow \mathrm{sample}(\pi(a \mid s))$

    $s', r \leftarrow G(s, a)$

    $\hat{u} \leftarrow \hat{u} + \gamma^t r$

    $s \leftarrow s'$

return $\hat{u}$

cf. Alg. 9.1, p. 184

Policy Evaluation

Naive Policy Evaluation not on Exam

Monte Carlo Policy Evaluation

- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*

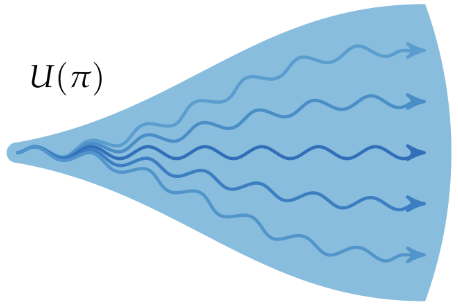
Let $\tau = (s_0, a_0, r_0, s_1, \ldots, s_T)$ be a *trajectory* of the MDP

$$U(\pi) \approx \frac{1}{m} \sum_{i=1}^{m} R(\tau^{(i)})$$

also an R.V.

$$U(\pi) \approx \bar{u}_m = \frac{1}{m} \sum_{i=1}^{m} \hat{u}^{(i)}$$

where $\hat{u}^{(i)}$ is generated by a rollout simulation



How can we quantify the accuracy of $\bar{u}_m$?

Value Function-Based Policy Evaluation

Guiding Questions

- What is a **Markov decision process**?
- What is a **policy**?
- How do we **evaluate** policies?

Break

- Suggest a policy that you think is optimal for the icy day problem

Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

MDP Example: Up-Down Problem

Dynamic Programming and Value Backup

Bellman's Principle of Optimality: Every sub-policy in an optimal policy is locally optimal

Policy Iteration

Algorithm: Policy Iteration

Given: MDP (S, A, R, T, γ)

1. initialize π, π' (differently)
2. while $\pi \neq \pi'$
3. $\quad \pi \leftarrow \pi'$
4. $\quad U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
5. $\quad \pi'(s) \leftarrow \underset{a \in A}{\operatorname{argmax}} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U^\pi(s') \right) \quad \forall s \in S$
6. return π

(Policy iteration notebook)

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

1. initialize U, U' (differently)
2. while $\|U - U'\|_\infty > \epsilon$
3. $\quad U \leftarrow U'$
4. $\quad U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U(s')\right) \qquad \forall s \in S$
5. return U'

- Returned U' will be close to U^*!
- π^* is easy to extract: $\pi^*(s) = \arg\max(R(s,a) + \gamma E[U^*(s)])$

Bellman's Equations

Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

"In any small change he will have to consider only these quantitative indices (or "values") in which all the relevant information is concentrated; and by adjusting the quantities one by one, he can appropriately rearrange his dispositions without having to solve the whole puzzle ab initio, or without needing at any stage to survey it at once in all its ramifications."

-- F. A. Hayek, "The use of knowledge in society", 1945